CLAIMS

1. A digital signal processor comprising:

an address generator configured to generate speculative data addresses in response to an address operand and one or more address parameters;

a pipelined execution unit configured to execute instructions using data at locations specified by the speculative data addresses;

a speculative register file configured to hold the speculative data addresses as corresponding instructions advance through the execution unit;

an architectural register file configured to hold architectural data addresses; and

control logic configured to write speculative data addresses to the speculative register file as the speculative data addresses are generated by the address generator and to supply speculative data addresses or architectural data addresses to the address generator.

2. A digital signal processor as defined in claim 1, wherein the speculative register file is configured with sufficient capacity to hold one or more architectural data addresses.

3. A digital signal processor as defined in claim 2, wherein the control logic is configured to move architectural data addresses from the speculative register file to the architectural register file in the event of a conflict for use of the speculative register file.

4. A digital signal processor as defined in claim 3, wherein the control logic is configured to write speculative data addresses to successive slots in the speculative register file.

5. A digital signal processor as defined in claim 4, wherein the control logic is configured to increment a pointer to a next available slot in the speculative register file.

6. A digital signal processor as defined in claim 5, wherein the control logic is configured to wrap the pointer from an end of the speculative register file to a start of the speculative register file.

7. A digital signal processor as defined in claim 3, wherein the control logic is configured to mark as architectural an entry in the speculative register file in response to the corresponding instruction being completed by the pipelined execution unit.

8. A digital signal processor as defined in claim 7, wherein the control logic is configured to mark as empty a slot in the speculative register file containing an old architectural data address when a current architectural data address is defined.

9. A digital signal processor as defined in claim 7, wherein the control logic is configured to mark as empty a slot in the speculative register file when the speculative data address stored therein does not become an architectural data address.

10. A digital signal processor as defined in claim 1, wherein the control logic is configured to update a control register corresponding to the one or more address parameters when a speculative data address is written to the speculative register file.

11. A digital signal processor as defined in claim 1, wherein the speculative register file comprises a circular buffer.

12. A digital signal processor as defined in claim 1, wherein the speculative register file has more slots than a number of pipeline stages in the pipelined execution unit.

13. A digital signal processor as defined in claim 1, wherein the speculative register file has two more slots than a number of stages in the pipelined execution unit.

14. A method for operating a digital signal processor, comprising:

generating a speculative data address in response to an address operand and one or more address parameters;

executing an instruction using data at a location specified by the speculative data address in a pipelined execution unit;

holding the speculative data address in a speculative register file as a corresponding instruction advances through the pipeline;

holding architectural data addresses in an architectural register file; and

writing the speculative data address to the speculative register file as the speculative data address is generated by the address generator.

15. A method as defined in claim 14, further comprising moving an architectural data address from the speculative register file to the architectural register file in the event of a conflict for use of the speculative register file.

16. A method as defined in claim 14, further comprising holding one or more architectural data addresses in the speculative register file.

17. A method as defined in claim 14, further comprising generating a next speculative data address based on a current speculative data address.

18. A method as defined in claim 14, further comprising marking as architectural an entry in the speculative register file when a corresponding instruction is completed by the pipelined execution unit.

19. A method as defined in claim 14, further comprising marking as empty a slot in the speculative register file containing an old architectural data address when a current architectural data address is defined.

20. A method as defined in claim 14, further comprising marking as empty a slot in the speculative register file when a speculative data address contained therein does not become an architectural data address.

21. A method as defined in claim 14, further comprising updating a control register corresponding to the one or more address parameters when the speculative data address is written to the speculative register file.